

2/4

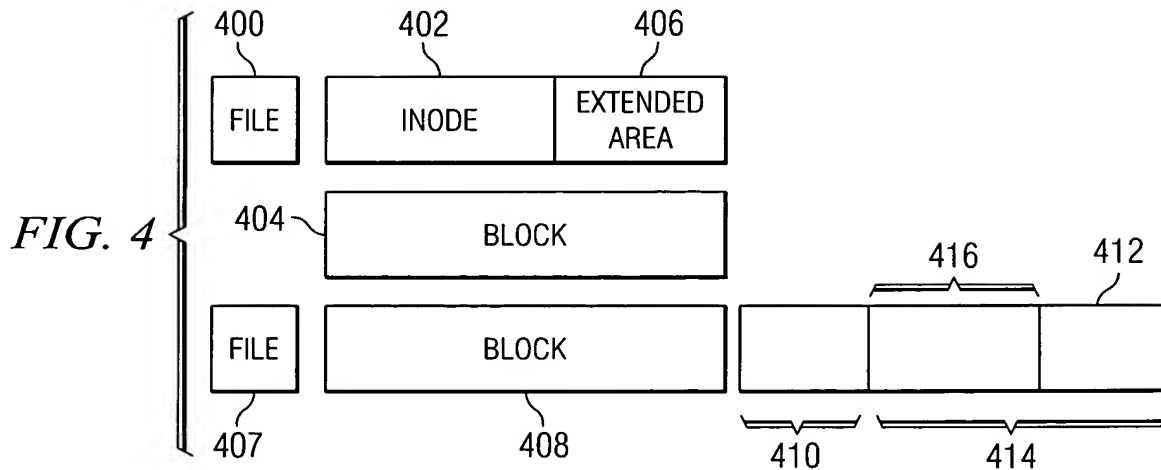


FIG. 5A

```

    500
    {
        /*
         *      I. base area (128 bytes)
         *      -----
         *
         *      define generic/POSIX attributes
         */
        inostamp; /* 4: stamp to show inode belongs to fileset */
        fileset; /* 4: fileset number */
        number; /* 4: inode number, aka file serial number */
        gen; /* 4: indode generation number */

        ixpdx; /* 8: inode extent descriptor */

        size; /* 8: size */
        502 nblocks; /* 8: number of blocks allocated */

        nlink; /* 4: number of links to the object */

        uid; /* 4: user id of owner */
        gid; /* 4: group id of owner */

        mode; /* 4: attribute, format and permission */

        atime; /* 8: time last data accessed */
        ctime; /* 8: time last status changed */
        mtime; /* 8: time last data modified */
        otime; /* 8: time created */

        di_acl; /* 16: acl descriptor */
        di_ea; /* 16: ea descriptor */
        504 { next_index; /* 4: Next available dir_table index */
    }
    
```

```

alctype; /* 4: Type of ACL */
/*
 * Extension Areas.
 *
 * Historically, the inode was partitioned into 4 128-byte areas,
 * the last 3 being defined as unions which could have mult
 * uses. The first 96 bytes had been completely unused until
 * an index table was added to the directory. It is now more
 * useful to describe the last 3/4 of the inode as a single
 * union.
 */
union {
    struct {
        /*
         * This table contains the information needed to
         * find a directory entry from a 32-bit index.
         * If the index is small enough, the table is in
         * otherwise, an x-tree root overlays this table
         */
        dir_table_slot_table[12]; /* 96: inline */
        _dtroot; /* 288: dtree root */
    } _dir; /* (384) */

    struct {
        union {
            data[96]; /* 96: unused */
            struct {
                _imap; /* 4: unused */
                _gengen; /* 4: generator */
            } _imap;
        } _ul; /* 96: */

        union {
            _xtroot[288];
            struct {
                unused[16]; /* 16: */
                _dxd; /* 16: */
                union {
                    dev; /* 4: */
                    _fastsymlink[128];
                } _u;
                _inlineea[128];
                _inlinedata[128];
            } _special;
        } _u2;
    } _file;
} u;

```

506

508

FIG. 5B

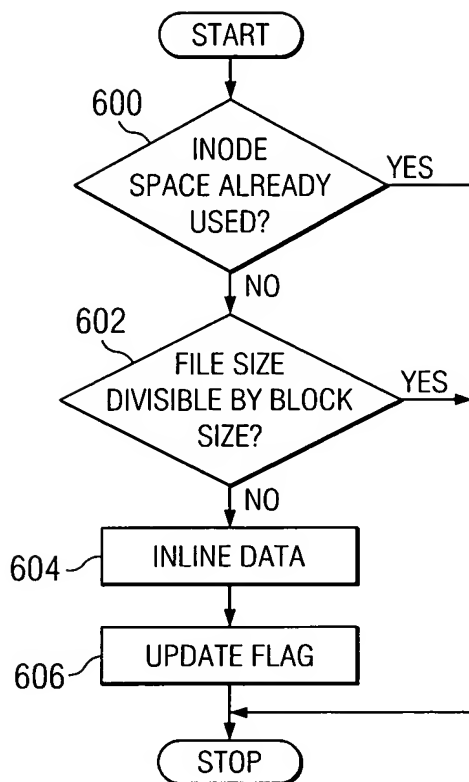


FIG. 6

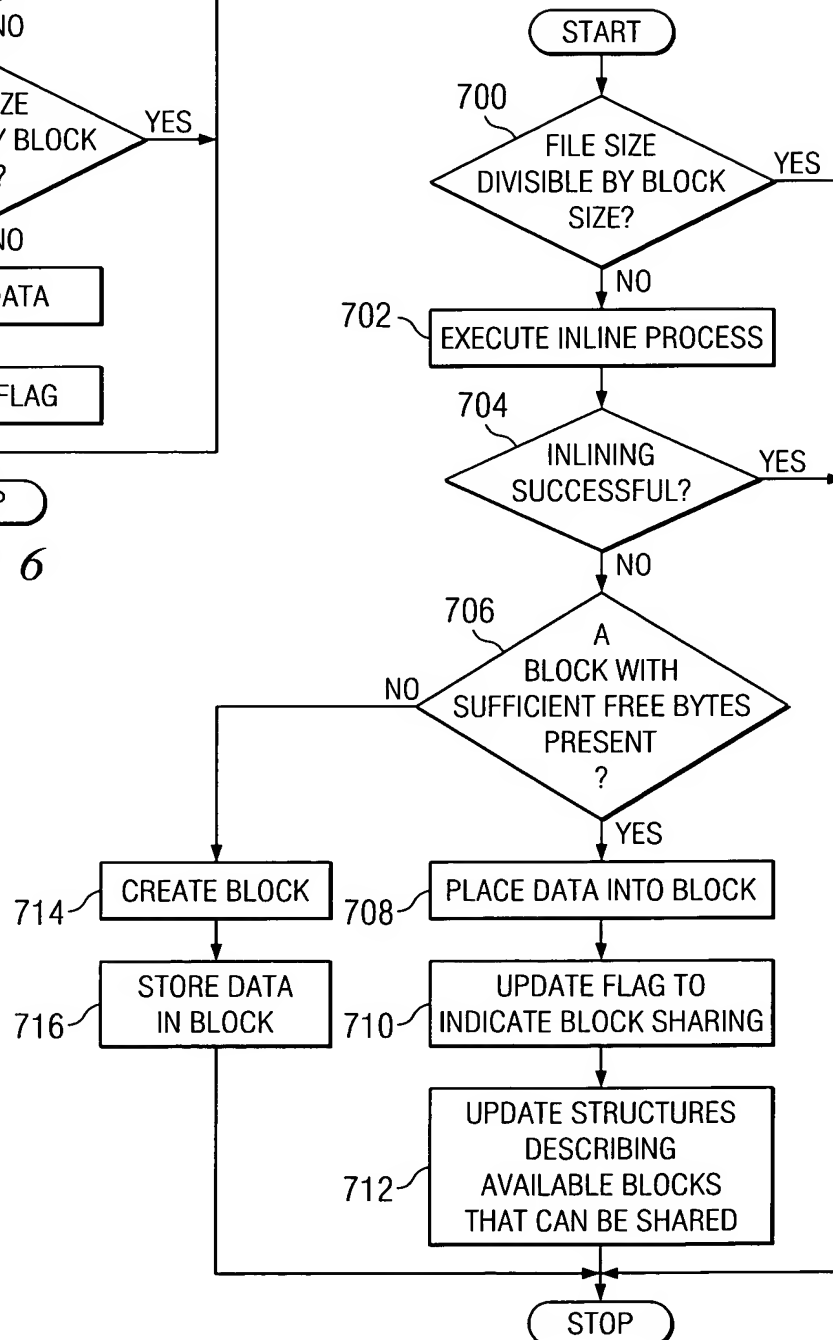


FIG. 7